Benchmarking of Cryptographic Algorithms

Alex Volkovitsky

January 27, 2005

Abstract

The author intends to validate theoretical numbers by constructing empirical sets of data on cryptographic algorithms. This data will then be used to give factual predictions on the security and efficiency of cryptography as it applies to modern day applications.

1 Introduction

Following is a description of the project and background information as researched by the author.

1.1 Background

1.1.1 Origins of Cryptography

Cryptography is a field of study into how two parties can exchange valuable information over an insecure channel. Historically speaking, the first use of encryption is attributed to Julius Caesar when he used a ROT(3) algorithm

to transfer military order within his empire. The algorithm was based on the simple premise of 'rotating' letters (hence the abbreviation ROT) by 3 characters such that 'a' became 'd', 'b' became 'e', etc. Decryption was the reverse of this process in that the receiving party needed merely to "un"-rotate the letters.

1.1.2 Basic Terminology and Concepts

At it's core cryptography assumes that two parties must communicate over some insecure channel. The sender (generally referred to as Alice) agrees on some encryption algorithm E(p,k) with the receiver on the other end (Bob). E(p,k) is generally some function of two or more variables, often mathematical in nature (such as all computer algorithms), but not necessarily (as was the case of the notorious Enigma machine used in World War II). The two variables in question are 'p', the plain-text or data that must get across without being read by any third party, and 'k', the key, some shared secret which both Alice and Bob have agreed to over a previously established secure connection. On the receiving end, Bob must possess a decryption function such that p=D(E(p,k),k). Meaning that if Bob knows the secret ('k'), he can retrieve the original message 'p'. The most important aspect of cryptography is the existence of the key which is able to transform seemingly random gibberish into valuable information.

1.1.3 Symmetric Algorithms

Symmetric key algorithms are algorithms which are most often used to transfer large amounts of data. Symmetric key algorithms use the same key 'k' to encrypt

and decrypt, and are generally based on relatively quick mathematic functions such as XOR. The downside of symmetric algorithms is the that since both parties must know the exact same key, that key needs to have been transfered securely in the past. This means that for Alice and Bob to communicate using a symmetric key algorithm, they must first either meet in person to exchange slips of paper with the key, or alternatively (as is done over the Internet) exchange a symmetric key over an established public/private-key connection. The most common modern symmetric algorithm is DES (Digital Encryption Standard).

1.1.4 Private/Public-Key Algorithms

Public-key cryptography is based on the concept that Alice and Bob do not share the same key. Generally, Alice would generate both the private key and the public key on her computer, save the private key and distribute the public key. If Bob would like to send a message to Alice, he first encrypts it with her public key, making her the only person able to decrypt the message. He sends the encrypted message (which even he himself can no longer decrypt) and Alice is able to read it using her private key. If Alice wishes to respond, she uses Bob's public key and follows a similar procedure. Alternatively, if Bob wishes to verify that it is Alice speaking and no one else, she can sign her messages. Signing is using your own private key to encrypt a message, such that anyone else may decrypt it and know that you were the only person who could've encrypted it. She would encrypt her message with her private key, then encrypt it with Bob's public key. Upon receiving the message he would be the only person able

to decrypt it (being the only person knowing Bob's private key) and then he would verify Alice's signature by decrypting the actual message with her public key. The most common modern day public-key algorithm is RSA, developed in 1977 by Ron Rivest, Adi Shamir and Len Adleman (hence the abbreviation RivestShamirAdleman, or RSA), which is based on a factoring problem.

1.2 Purpose of the Project

Whereas much research has been done into theoretical cryptography, very little has been done to prove simple formula numbers and to look into the speeds at which various algorithms operate. My project seeks to observe several modern day algorithms and to compute empirical data on the time it takes to encrypt and/or decrypt different amounts of data, and how different algorithms perform with varied key lengths, modes of operation, and data sizes. Ideally my program could be run on different types of machines to identify if certain architectures give an advantage to the repeating mathematical computations required by cryptographic algorithms. My project also seeks to try to break several algorithms using unrealistically small key lengths (using real key lengths such as 64-bit could take years to break using brute-force methods), this way I could extrapolate my data and give predictions on the security afforded by actual key lengths.

1.3 Scope

2 Development

3 Results

4 Conclusions

5 Summary

6 References

1. "Handbook of Applied Cryptography" University of Waterloo. 27 Jan. 2005. <http://www.cacr.math.uwaterloo.ca/hac/>.

2. "MCrypt" Sourceforge. 27 Jan. 2005. <http://mcrypt.sourceforge.net>.

3. "Cryptography FAQ." sci.crypt newsgroup. 27 June 1999. <http://www.faqs.org/faqs/cryptography-faq/>.